

The first chip dedicated to text manipulation, the 82730 operates as a coprocessor to a host CPU and executes many high-level commands that reduce the software needed for processing text.

Text coprocessor brings quality to CRT displays

The quality of text in raster-scanning CRT displays has always been a tradeoff against the complexity, performance, and cost of the associated video system. By allocating many of the complex display functions to firmware, a dedicated text coprocessor chip, the first of its kind, replaces printed-circuit boards that contain more than 100 ICs while increasing system performance by relieving many of the host processor's text manipulation tasks. The chip thus makes possible the high quality and high performance sought, without the need to compromise because of high design complexity and high cost of text-processing hardware.

Though its speed makes the 82730 text coprocessor beneficial on its own, its utility can be enhanced considerably when working with the 82731 video interface controller. Together they provide proportional spacing, simultaneous subscript and superscript displays, dual cursors, dynamically reloadable character fonts, and user-programmable field and character attributes. By adding still an-



other chip, the 82720 graphics display controller, the device can display high-resolution graphics and text at the same time.

Housed in a 68-pin package, the 82730 text coprocessor combines a direct memory access channel and a processor bus interface that permit it to fetch its own instructions and data from the host system's memory, independent of and in parallel with the host CPU.

The two processors communicate through messages—commands, parameters, and status words—which are placed in a communication block inside a shared memory. The

host issues commands by preparing messages, storing them in the communication block, and directing the coprocessor's attention to them by activating a Channel Attention signal, which is implemented in hardware. In return, the coprocessor sets a flag in the shared memory that notifies the host when it has executed the command.

The 29 high-level commands built into the 82730 break down into two groups: channel commands, which work at the system level to start and stop the display and to communicate status and similar information, and data-stream commands, which are incorporated directly into the display-data strings to govern the DMA process and control row

Anand Balaram, Product Marketing Engineer
Andrew Volk, Project Manager
Intel Corp.
3065 Bowers Ave., Santa Clara, Calif. 95051

Text coprocessor

characteristics, character attributes, and so on.

The 82730 resides on a local system bus with the host microprocessor, such as the 80186 CPU, and therefore provides the same address, data, and control signals as the main processor. By handling several of the tasks typically done by the host processor—like DMA control and display formatting—it leaves the host free for other tasks.

For example, when the coprocessor is configured to share the CPU bus, a portion of the host microprocessor bus bandwidth must be devoted to the DMA process that refreshes the CRT. However, the 82730's high-speed intelligent DMA controller (operating at a maximum data rate of 4 Mbytes/s) helps minimize the time spent executing the refresh operation, while simultaneously handling the formatting of the display data. A different approach involves a dual-ported memory architecture, which places the memory between the CPU and the coprocessor. That completely frees the processor bus of the refresh activity, allowing the host more time to execute other tasks. It has become a more cost-effective method, as some dynamic memory controllers now contain dual-ported arbitration logic on chip.

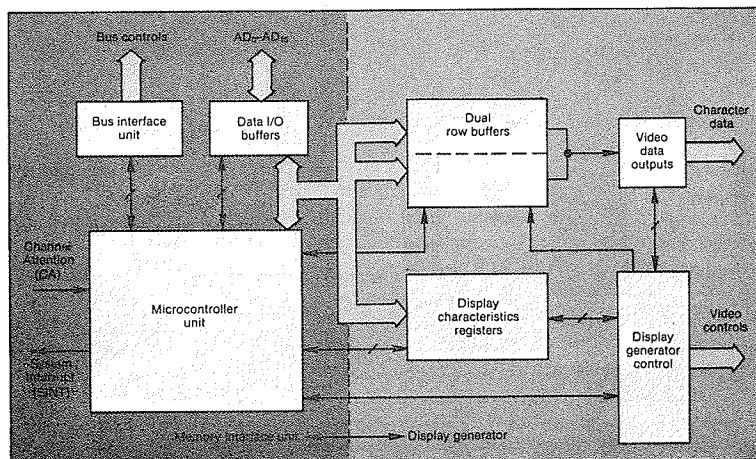
Inside the chip

The basic architecture of the coprocessor is divided into two main parts: a memory interface and a display generator section (Fig. 1). The memory interface lets the coprocessor and the system pro-

cessor communicate via the shared memory. The display generator, in turn, responds to the data provided by the memory interface and carries out the display operations.

The memory interface actually comprises two smaller subsections, a bus interface unit and a microcontroller unit. The bus interface provides an intelligent connection from the 82730 to the host processor bus and also buffers the data transfer requests from the microcontroller. Upon initialization, the bus interface can be programmed for 8- or 16-bit data and 16- or 32-bit addresses. Furthermore, the host interface can be configured for 8- or 16-bit-wide data buses, making the coprocessor compatible with 8- or 16-bit host processors, like the 8088/80188 and the 8086/80186. Running at 8 MHz maximum in 16-bit systems, the 82730 handles the maximum DMA rate of 4 Mbytes/s.

The microcontroller unit stores the microinstructions for the 82730's high-level operations. The microcontroller's internal processor manages the memory transfers, interprets the commands embedded in the data stream, and executes those commands by sending data to the appropriate control registers or display data buffers. To optimize the transfer of data between the system and the CRT interface, the coprocessor uses three clocks—one for the host interface, the other two for video data. The memory interface section runs from the bus clock, the CRT interface operates from a reference and a character clock.



1. Divided into two main sections—a memory interface unit and a display generator—the 82730 text coprocessor can operate at optimum speed since each section can function independently at a different clock speed.

Although the coprocessor packs a considerable amount of processing power on a single NMOS chip, it cannot handle the high video dot rate needed to deliver high character counts to the CRT display. For that, it needs the 82731 video interface controller, which gains its high speed and drive capability from bipolar technology. In addition, the combination of the 82730 and 82731 succeeds in reducing the video interface to just a few latches and a software character generator residing in RAM or ROM (Fig. 2).

Inside the 82731 are the reference- and character-clock generators, a video shift register, and all attribute logic (Fig. 3). Housed in a 40-pin package, the circuit offers TTL-compatible inputs and outputs except for the video output, which is ECL-compatible and provides a dot-shift clock rate of 50 MHz maximum on characters up to 16 dots wide. The circuit proportionally spaces characters by accepting the width sent from the character generator and sending an appropriate character-clock output whose period determines the variable width of the character to be displayed.

The video interface controller can employ an inexpensive, low-frequency crystal and internally multiply that frequency to generate the high-frequency dot clock. It also supports control functions such as screen reverse video, synchronized character field, and tabbing operations. The dot clock drives the internal video shift register, the character clock controls the unloading of data from

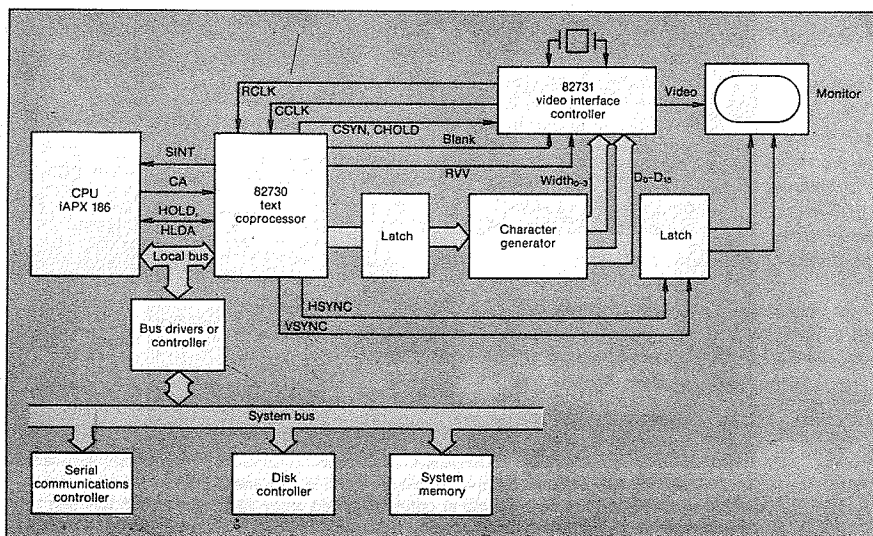
the row buffers in the 82730, and the reference clock establishes the raster and screen formats. The reference clock also supplies the basic timing for the horizontal sync, blanking, border, and active display time. The corresponding vertical attributes—except border—are driven by the horizontal line time. All seven of these screen parameters are programmable by the system designer with the 82730.

System interfaces are simple

As a coprocessor, the 82730 has the same bus-control signals as an 80186 host processor and thus can share the system-bus controllers, drivers, and latches. The host processor and the 82730 arbitrate for control of the local bus through the Hold and Hold Acknowledge lines (HLD/HLDA). The Channel Attention (CA) and System Interrupt (SINT) control lines complete the wired interface. With this configuration, the 82730 has access to all the memory that the 80186 CPU has available.

Anytime the CPU wants to send a message to the 82730, it writes the command and busy flag into the 82730 command block and then pulses the coprocessor's CA input to inform it that a message is waiting. The 82730 then raises the HOLD output and waits for access to the bus. When the CPU relinquishes the bus, it raises the HLDA input of the 82730.

Once the 82730 becomes active, it transmits the command block address that was stored in its



2. A typical system built around the 82730 and the 82731 video interface controller requires very few additional ICs to mate with a host processor like the 80186. Only the system bus drivers, some latches, and a character generator are needed.

Text coprocessor

registers during initialization. That address, in conjunction with the appropriate memory control signals—including read or write strobes, lower or upper address latch enables, upper address output, or data enable output—will either read the command block or write to it. All these signals are coordinated by the bus clock.

Whenever the 82730 must send status information to the host CPU, it gains control of the bus and places the data into the status location in the command block. The bus is then released and the coprocessor notifies the CPU through the SINT signal. When the coprocessor is using a dual-ported memory to communicate with the 82730, the HOLD and HLDA signals are not employed. Instead, the 82730 accesses the dual-ported memory directly rather than acquiring the bus from the CPU.

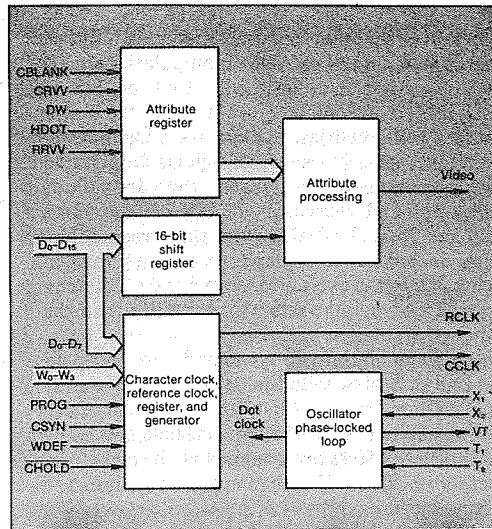
When the display process is activated, the coprocessor uses its built-in DMA capability to fetch display data from the memory. The data consists of character data mixed with data-stream commands; embedded data-stream commands provide the flexibility to manipulate data on the fly.

Soft fonts loaded

The 82730 also permits soft fonts to be automatically loaded into RAM-based character generators. Addresses and data stored in the system memory are then loaded into the row buffers of the coprocessor. During blanked rows (generally during the vertical retrace), address information is loaded into a latch and data is written to the character generator.

The 82730's dual row buffers help reduce the bandwidth and access time requirements for the system memory. The data stored in one buffer is being used to display a row on the screen while the second buffer is being loaded, by the microcontroller, with the next display row from the system memory. Up to 200 characters can be stored and displayed by each row buffer. Furthermore, since the display generator section operates asynchronously with the microcontroller unit, each can operate at optimal speed. Processing is synchronized by internal flags and shared internal storage, and data that will be displayed is exchanged through the row buffers.

The coprocessor's display generator handles the data that defines the timing and the operation of the CRT interface. That data, which is stored in the display characteristics registers of the chip, controls every aspect of the display—from the screen's format to the blink rates of the characters and cursors. All the parameters that define the initial display characteristics can be set by one command—MODEST—thus reducing the time and



3. The 82731 video interface controller is manufactured with bipolar technology, enabling it to handle video dot rates of 50 MHz and higher, which are needed by high-character-count displays. The controller serializes the parallel character outputs from the coprocessor and adds the desired attributes to each character.

effort required to establish a screen format.

Beneath the simplicity of the hardware shown in Fig. 2 are the high-level instructions—channel commands—and the data-stream commands. When combined with a table-driven linked-list data structure, they ease software development.

Central to the software is the command block, through which all channel commands are transferred between the coprocessor and the host. This block is located within the shared memory, and its exact position is set during the 82730's initialization routine (Fig. 3a). Once established, it contains all the information needed to start the display-data fetch; to communicate status, interrupt, and cursor position information; and to give the location of the mode block, which contains all the parameters for setting up the display. The START DISPLAY channel command begins the sequence (Fig. 3b).

Since the display data is set up within linked lists, the coprocessor can rapidly change any of the lists without shifting huge amounts of data. The display fetch starts with the value of the list-switch bit which selects one of two list-base pointers in the command block. The pointer points to its string pointer list; the pointers in that list direct the on-chip DMA to the data strings containing the desired display data and data-stream commands. The programmer can modify one pointer list while

displaying from the other, and can also switch screens merely by changing the list-switch bit, thus eliminating time-critical data manipulations.

Two data-stream commands—End of String (EOS) and End of Row (EOR)—are key to the linked list and DMA activities. Strings are a logical concept: they contain blocks of contiguous data stored anywhere in memory. In contrast, rows are a physical concept and represent a block of characters that make up a physical row on the screen. Many strings can exist in a display row, or many rows in a string. (Only the extra DMA overhead of fetching the new string pointer sets a practical limit on the number of strings in each row.)

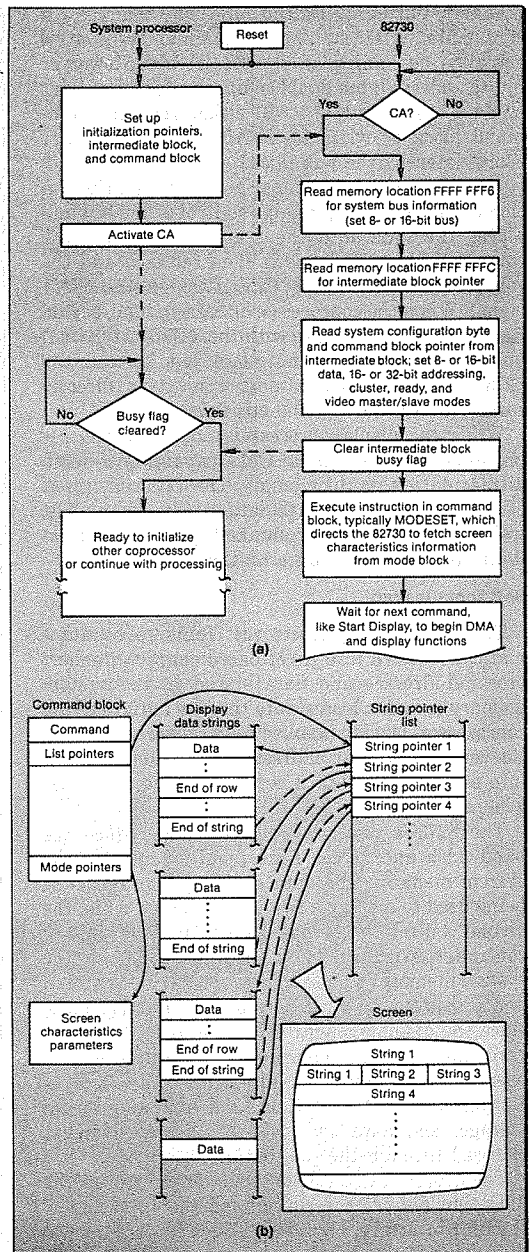
The actions of the two commands are independent. End of String tells the 82730 to get the next string pointer from the list, and from there, the next data string. End of Row suspends the DMA until the row buffers are swapped at the end of the current row. The DMA then takes over, into the new row buffer.

String manipulation fosters high speed

Strings are commonly the next level of text organization above single characters. With the 82730, a string can be as small as a character or it can be a word, row, sentence, paragraph, or a page of characters. These high-level entities can be moved merely by manipulating a small string pointer table (Fig. 5). The heart of the algorithm for word wraparound, a common feature in text processors, can easily be accommodated by a single command such as the String Compare command of the 80186. Word wraparound is then achieved by scanning the data (not moving it) and manipulating a few pointers. Earlier system designs would have required a multiple-instruction software loop that scanned and moved every individual character.

An extension of the linked list allows programmers to set up several independent data windows on the CRT screen in a virtual screen mode. That feature is especially helpful if a user wants both a menu window and one or more work-space windows. Such a scheme saves a lot of time for the end user—eliminating the back-and-forth movement between menus and working text. To set this up, several data structures, each with its own command block, can be accessed in a table-driven sequence to put data in a given window on the screen (Fig. 6).

The string list and data strings are the same for regular or virtual modes; only the structure of their command blocks differs. Thus, each virtual window can be an independent software entity in the system, and the 82730 can present these independent data bases simultaneously.



4. Both the host CPU and the coprocessor go through an initialization sequence when the computer system is reset (a). The coprocessor then looks for a START DISPLAY command so that it can load the various data strings from the system memory into the display generator section, attach attributes, and display the data on the CRT (b).

Text coprocessor

Multiple 82730s can also be used in a single system. Up to four devices can be clustered in a single system, with one serving as a system master and the others as slaves. The data for this cluster can be interleaved, permitting the cluster to work from one data base to get more characters per screen or more bits per character. Also, in the slave mode, the 82730's video outputs can be synchronized to an external video signal, giving the system such capabilities as mixed text and graphics, broadcast subtitling (text overlay), and overlays for video recording.

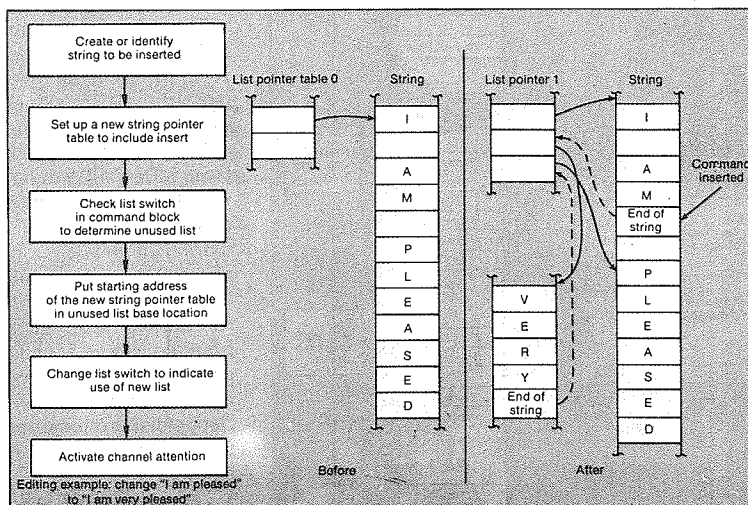
Attributes enhance display quality

The designers of the 82730 have given it the ability to highlight various areas of an on-screen document through the use of character and field attributes. In the 16-bit data word, for example, only the most significant bit is committed; it serves as the command or data designator. If set to 1, the word is a data-stream command, with the remaining 15 bits becoming one of the predefined instructions. However, if the MSB is 0, the other bits are at the discretion of the designer, who may choose which and how many are needed for charac-

ter codes, attributes, or user-defined functions.

The 82730's six predefined attributes—reverse video, invisible, blinking character, two underlines, and a special graphics character—can be programmed to respond to any of the 15 bits, or they can be completely disabled. In addition, they can be set character by character or through a field-attribute mask. All can be attached to any character. The blinking character can be programmed for a wide range of duty cycles and blink rates. The two underlines can be independently positioned anywhere in the row height, and the position can be changed from row to row. Thus the underline can be doubled or serve as a strike-through line, a fraction line, or an overbar. One of the underlines can also be programmed to blink at the same rate as a blinking character.

The graphics character is relatively important, since it permits character information to be displayed to the full height of the row. It causes the chip's line-counter output to count from zero at the top of the display row continuously through to the bottom of the row. When used with special characters, this attribute allows business forms and graphs to be easily constructed.



5. If a character or word must be inserted near the beginning of a screen of text, only the list pointers must be changed to add the item. In older systems, all the characters following the insertion or deletion were shifted in the memory to revise the display.

Text coprocessor

Another capability of the 82730 is subscript and superscript characters, done by manipulating the line-counter outputs. The SUB SUP N data-stream command declares which and how many pairs of characters will be displayed simultaneously as subscripts and superscripts.

Proportionally spaced displays could cause some subscript and superscript characters to have different widths and thus disrupt the vertical alignment of a character pair. A special output of the 82730 called Width Defeat prevents that misalignment by causing the 82731 video interface controller to enforce a predefined width—programmed upon system initialization—during the display of subscript and superscript characters.

The proportional spacing is performed by the reference and the character clock. Used to shift out the character and attribute data, the character clock operates during the display field. Its frequency can vary character by character, up to a rate of 10 MHz, to set the width of the character currently being displayed. The video interface controller takes the character width information that has been supplied by the character generator and

produces a variable width character clock that supports the proportional spacing. This approach also greatly reduces system complexity and cost compared with previous designs.

Screen and row formats are flexible

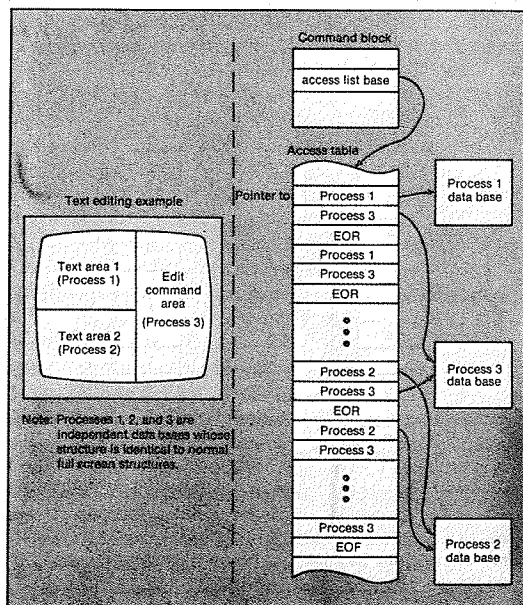
The reference clock signal in a system that contains the 82730 and 82731 chips is a constant-frequency clock that forms the time base to generate the horizontal scan lines and vertical frame periods. One scan line can last for 256 reference clock periods, and one frame can contain up to 2048 scan lines. Within these periods, the respective synchronization pulses and the border and character fields can be set anywhere within that range. All these timing relationships, including the scan and frame periods, can be changed on a frame-by-frame basis to suit changing applications.

The screen format is flexible all the way down to the row level. For instance, the height of a row (up to 32 scan lines) and the vertical position of the characters within that row can be changed from row to row by a single data-stream command called FULROWDESCRPT. In addition, the command lets the programmer set the starting and ending scan lines within the row for the normal, subscript, and superscript character fields and the two cursors.

The same data-stream command that defines the row characteristics can also be used to blank the row, display it as reverse video, double its height (for up to 64 scan lines per row), or eliminate the proportional spacing.

Graphics, too, can be handled by the 82730, although flexibility and resolution are not as high as with the 82720 graphics display processor. Business applications typically need graphics that are no more complex than two- or three-dimensional charts or business forms. These formats can be stored as special characters in a standard font set for the character generator. Even more complex graphics can be handled through the use of mosaic graphic cells, which can be stored in RAM to permit alterations. Of course, as in most systems using floppy-disk systems for main storage, the desired fonts or graphics forms can be saved on the disks and downloaded as needed for the display.

There are many applications that also require a simple graphic display along with text—signature verification on banking terminals and general-purpose credit verification, for example.□



6. The virtual window capability of the 82730 lets the user arrange independent areas in the system memory that can be displayed simultaneously on the CRT monitor.